

**DEPARTMENT OF**

**COMPUTER SCIENCE AND ENGINEERING**

**To-Do List Application**

UI/UX Design Fundamental

B. Tech Degree – 3 BTCS Section B

**School of Engineering and Technology,**

**CHRIST (Deemed to be University),**

**Kumbalagodu, Bengaluru-560 074**

September 2025



***Certificate***

*This is to certify that K Chris Kevin A(2460349), Kevin Patrick K(2460419), Kavin M(2460391) has successfully completed the Project work for UI/UX Design Fundamental in partial fulfillment for the award of Bachelor of Technology during the academic year 2025-2026.*

*Mrs Nagaveena*

**FACULTY- IN CHARGE**

Name : Chris Kevin A.Patrick Kevin k, Kavin M

Register No. : : 2460349,2460419,2460391

E-mail id : chris.kevina@btech.christuniversity.in , [Patrick.kevin@btech.christuniversity.in](mailto:Patrick.kevin@btech.christuniversity.in), kavin.m@btech.christuniversity.in

Date :25/09/2025

Institution :SoET, Christ University(Kengeri Campus)

# INDEX

|  |  |
| --- | --- |
| **SI. No** | **Contents** |
| **1** | **Abstract** |
| **2** | **Objective** |
| **3** | **Scope of Project** |
| **4** | **Tools and Technologies used** |
| **5** | **HTML Structure Overview** |
| **6** | **CSS Styling Strategy** |
| **7** | **Key Features** |
| **8** | **Challenges Faced & Solutions** |
| **9** | **Outcome** |
| **10** | **Future Enhancement** |
| **11** | **Sample Code** |
| **12** | **Final Output** |
| **13** | **Conclusion** |
| **14** | **References** |

## 1.Abstract

The To-Do List App is a simple yet effective task management web application that allows users to add, mark, filter, and manage daily tasks. It is built with **HTML, CSS, JavaScript, Bootstrap, and jQuery**, ensuring both functionality and responsive design. The app helps users stay organized and productive by providing features like filtering tasks (all, active, completed) and a clean, user-friendly interface.

### 2. Objective

The primary objectives of this project are:

* To create a functional and interactive task management application.
* To practice DOM manipulation using JavaScript/jQuery.
* To design a simple, responsive, and user-friendly UI with Bootstrap.
* To help users manage their daily tasks effectively.

### 3. Scope of Project

The scope of this project goes beyond just displaying images.

* Can be used by students, professionals, and individuals for task organization.
* Provides essential CRUD (Create, Read, Update, Delete) operations for tasks.
* Can be extended into a larger productivity app with deadlines, notifications, or cloud storage.

### 4. Tools and Technologies Used

* HTML5 → Structure of the app.
* CSS3 → Styling and layout.
* Bootstrap 5 → Responsive design and pre-built UI components.
* JavaScript + jQuery → DOM manipulation, event handling, filtering logic.
* Google Fonts (Poppins) → Modern typography.

### 5. HTML Structure Overview

* **Task Input Section → Input box + Add button.**
* **Filter Buttons → All, Active, Completed.**
* **Task List Section → Dynamic list of tasks (ul/li).**
* **Task Input Section** → Input box + Add button.
* **Filter Buttons** → All, Active, Completed.
* **Task List Section** → Dynamic list of tasks (ul/li).

### 6. CSS Styling Strategy

* Used Bootstrap utility classes for layout and spacing.
* Applied custom CSS for task highlighting, completed task strike-through, and hover effects.
* Used consistent font (Poppins) for a modern look.

### 7. Key Features

### Add tasks dynamically.

### Mark tasks as completed.

### Delete tasks from the list.

### Filter tasks (all, active, completed).

### Responsive design for mobile and desktop.

### 8. Challenges Faced & Solutions

* Challenge: Dynamically updating task list on user actions.
* *Solution:* Used jQuery to append, remove, and toggle classes.
* Challenge: Filtering tasks efficiently.
* *Solution:* Added data-filter attributes and jQuery logic to show/hide tasks.
* Challenge: Keeping UI simple and intuitive.
* *Solution:* Used Bootstrap’s grid system and buttons for clean layout.

### 9. Outcome

The To-Do List App successfully allows users to manage their tasks. It is responsive, easy to use, and can be extended into a more advanced productivity tool.This project demonstrates how front-end technologies can be combined to create **professional-quality web experiences**.

### 10. Future Enhancement

* Store tasks using LocalStorage so data persists after page reload.
* Add task deadlines and reminders.
* Include a dark mode toggle.
* Add task categories (work, personal, study).
* Sync tasks with a backend or cloud database.

### 11. Sample Code

**index.html**

project number 2:

index

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>To-Do List App</title>

  <!-- Bootstrap -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

  <!-- Custom CSS -->

  <link rel="stylesheet" href="style.css">

  <!-- Google Font -->

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap" rel="stylesheet">

</head>

<body class="bg-light">

  <div class="container py-5">

    <h1 class="text-center mb-4 fw-bold">📝 To-Do List</h1>

    <!-- Task Input -->

    <div class="input-group mb-4">

      <input type="text" id="taskInput" class="form-control" placeholder="Enter new task">

      <button class="btn btn-primary" id="addTaskBtn">Add Task</button>

    </div>

    <!-- Filter Options -->

    <div class="mb-3 text-center">

      <button class="btn btn-outline-secondary filter-btn active" data-filter="all">All</button>

      <button class="btn btn-outline-secondary filter-btn" data-filter="active">Active</button>

      <button class="btn btn-outline-secondary filter-btn" data-filter="completed">Completed</button>

    </div>

    <!-- Task List -->

    <ul id="taskList" class="list-group">

      <!-- Tasks will be added dynamically -->

    </ul>

  </div>

  <!-- Bootstrap + jQuery -->

  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

  <script src="script.js"></script>

</body>

</html>

**Style.css**

style

body {

  font-family: 'Poppins', sans-serif;

}

.completed {

  text-decoration: line-through;

  opacity: 0.6;

}

.list-group-item {

  display: flex;

  align-items: center;

  justify-content: space-between;

}

.task-actions button {

  margin-left: 5px;

}

**Script.js**

script

$(document).ready(function () {

  // Add Task

  $("#addTaskBtn").click(function () {

    let taskText = $("#taskInput").val().trim();

    if (taskText !== "") {

      addTask(taskText);

      $("#taskInput").val("");

    }

  });

  // Add Task on Enter key

  $("#taskInput").keypress(function (e) {

    if (e.which === 13) {

      $("#addTaskBtn").click();

    }

  });

  // Function to create a new task

  function addTask(text) {

    let task = `

      <li class="list-group-item">

        <div class="form-check">

          <input class="form-check-input task-check" type="checkbox">

          <label class="form-check-label">${text}</label>

        </div>

        <div class="task-actions">

          <button class="btn btn-sm btn-warning edit-btn">Edit</button>

          <button class="btn btn-sm btn-danger delete-btn">Delete</button>

        </div>

      </li>

    `;

    $("#taskList").append(task);

  }

  // Toggle Completed

  $(document).on("change", ".task-check", function () {

    $(this).closest("li").find("label").toggleClass("completed");

  });

  // Delete Task

  $(document).on("click", ".delete-btn", function () {

    $(this).closest("li").remove();

  });

  // Edit Task

  $(document).on("click", ".edit-btn", function () {

    let label = $(this).closest("li").find("label");

    let currentText = label.text();

    let newText = prompt("Edit task:", currentText);

    if (newText !== null && newText.trim() !== "") {

      label.text(newText.trim());

    }

  });

  // Filter Tasks

  $(".filter-btn").click(function () {

    $(".filter-btn").removeClass("active");

    $(this).addClass("active");

    let filter = $(this).data("filter");

    $("#taskList li").each(function () {

      let isChecked = $(this).find(".task-check").is(":checked");

      if (filter === "all") {

        $(this).show();

      } else if (filter === "active" && !isChecked) {

        $(this).show();

      } else if (filter === "completed" && isChecked) {

        $(this).show();

      } else {

        $(this).hide();

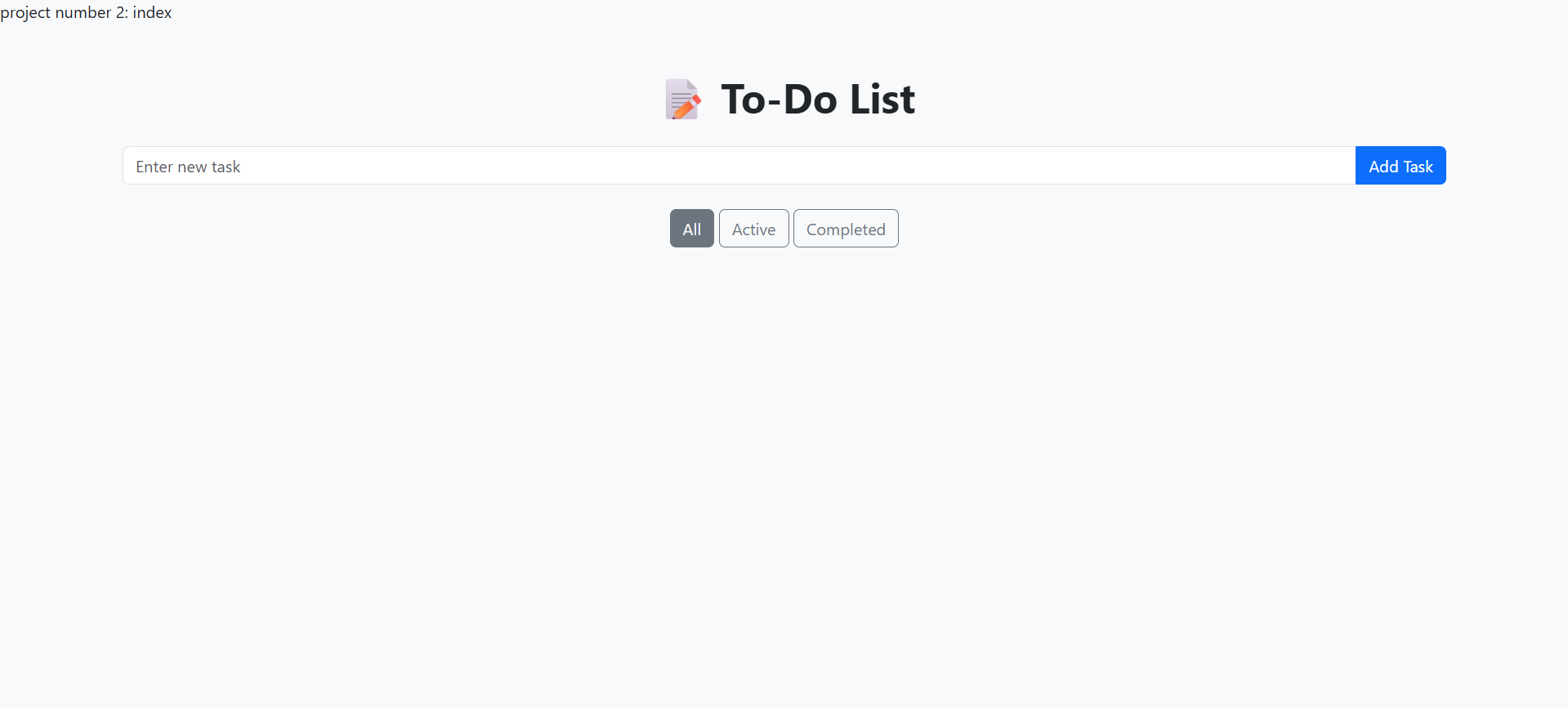
      }

    });

  });

});

### 12. Final Output



#### 13. Conclusion

The To-Do List App demonstrates practical usage of front-end technologies to build an interactive, responsive, and useful productivity tool. It highlights concepts of DOM manipulation, event handling, and UI/UX design.

#### 14. References

* Bootstrap Documentation
* jQuery Documentation
* MDN Web Docs
* W3Schools
* [Unsplash](https://unsplash.com/) – Stock images
* Wallpapers.com – Backgrounds